WHAT IS CLAIMED IS:

1. A method of generating object-oriented computer programs for accessing and updating persistently stored objects, comprising the steps of:

receiving an initial computer program that includes original instructions for accessing and updating objects stored in a computer's main memory;

automatically revising said initial computer program to generate a revised computer program by:

adding object loading instructions that, during execution of said revised computer program, load respective ones of said objects from persistent storage into said main memory when each said respective object is accessed for a first time; and

adding object storing instructions that, during execution of said revised computer program, store respective ones of said objects in said computer's main memory into said persistent storage upon the occurrence of predefined events.

2. The method of claim 1,

said revising step further including:

adding dirty object marking instructions that, during execution of said revised computer program, store data indicating which objects in said computer's main memory contain new and/or updated data;

wherein said object storing instructions store in said persistent storage those of said objects in said computer's main memory that contain new and/or updated data.

3. The method of claim 1, wherein said initial computer program is a compiled, object code or bytecode program.

4. The method of claim 3, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said

persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said revising step further including:

adding supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

5. The method of claim 4, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said revising step includes revising said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.

6. The method of claim 1, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said revising step further including:

adding supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main

memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

7. The method of claim 6, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said revising step includes revising said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.

8. A memory for storing data for access by programs being executed on a data processing system, said memory comprising:

a postprocessor procedure for modifying an initial computer program that includes original instructions for accessing and updating objects stored in a computer's main memory;

said postprocessor procedure including instructions for automatically revising said initial computer program to generate a revised computer program by adding supplemental instructions to said initial computer program, said supplemental instructions including:

object loading instructions that, during execution of said revised computer program, load respective ones of said objects from persistent storage into said main memory when each said respective object is accessed for a first time; and

object storing instructions that, during execution of said revised computer program, store respective ones of said objects in said computer's main memory into said persistent storage upon the occurrence of predefined events.

9. The memory of claim 8,

said supplemental instructions including:

dirty object marking instructions that, during execution of said revised computer program, store data indicating which objects in said computer's main memory contain new and/or updated data;

wherein said object storing instructions store in said persistent storage those of said objects in said computer's main memory that contain new and/or updated data.

10. The memory of claim 8, wherein said initial computer program is a compiled, object code or bytecode program.

11. The memory of claim 10, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said supplemental instructions further including:

supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

12. The memory of claim 11, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said postprocessor procedure revises said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.

13. The memory of claim 8, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said supplemental instructions further including:

supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

14. The memory of claim 13, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said postprocessor procedure revises said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.

15. A computer system, comprising:

memory, including a main memory for storing objects;

said memory further storing an initial computer program, a revised computer program and a postprocessor procedure for modifying an initial computer program that includes original instructions for accessing and updating objects stored in said main memory;

said postprocessor procedure including instructions for automatically revising said initial computer program to generate a revised computer

program by adding supplemental instructions to said initial computer program, said supplemental instructions including:

object loading instructions that, during execution of said revised computer program, load respective ones of said objects from persistent storage into said main memory when each said respective object is accessed for a first time; and

object storing instructions that, during execution of said revised computer program, store respective ones of said objects in said computer's main memory into said persistent storage upon the occurrence of predefined events.

16. The computer system of claim 15,

said supplemental instructions including:

dirty object marking instructions that, during execution of said revised computer program, store data indicating which objects in said computer's main memory contain new and/or updated data;

wherein said object storing instructions store in said persistent storage those of said objects in said computer's main memory that contain new and/or updated data.

17. The computer system of claim 14, wherein said initial computer program is a compiled, object code or bytecode program.

18. The computer system of claim 17, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said supplemental instructions further including:

supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

19. The computer system of claim 18, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said postprocessor procedure revises said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.

20. The computer system of claim 15, wherein

said objects accessed and updated by said original instructions include objects in a first set of object classes, and said objects stored in said persistent storage by said object storing instructions include objects in a second set of object classes that is a subset of said first set;

said original instructions including object data structure defining instructions for defining data structures associated with said objects in said first set of object classes;

said supplemental instructions further including:

supplemental object definition instructions that revise said data structures associated with said second set of object classes so as to enable said objects in said second set of object classes to store both main memory object pointers and persistent storage object identifiers for objects referenced by said objects in said second set of object classes.

21. The memory of claim 20, wherein

objects loaded into main memory by said object loading instructions include null object pointers for objects in persistent storage that are referenced by said objects loaded in main memory;

said postprocessor procedure revises said initial computer program so that said object loading instructions are invoked whenever a null object pointer in an object in said second set of object classes is accessed.